

## REMARKS

This Amendment is in response to the Office Action mailed September 13, 2007.

In the Office Action, claim 71 is withdrawn herewith, as being dependent upon a previously withdrawn claim. This withdrawal obviates the Office's objection under 35 C.F.R. §1.75(c) and rejection under 35 U.S.C. §112(2).

Claim 4 has been amended to refer to the inbound payload, which finds antecedent basis in claim 1.

Claims 1-12, 63-65, 71 and 78-110, were rejected under 35 U.S.C. §103(a) as being unpatentable over Mockapetris in view of Nguyen. Reconsideration and withdrawal of these rejections are hereby respectfully requested.

### Independent Claim 1 and Its Dependent Claims

Claim 1, as amended recites:

at least one gaming machine coupled to the at least two central servers through the communication network in a client-server configuration in which each of the at least one gaming machine is a client to the at least two central servers, each of the at least one gaming machine being configured to play at least one game and to carry out a game transaction for each game played and to commit each game transaction to each of the at least two central servers by sending a separate transaction packet to each of the at least two central servers, each of the separate transaction packets sent to each of the at least two central servers including an identical inbound game payload wherein each of the at least two central servers, upon receipt of the inbound game payload, are configured to return an outbound game payload to the gaming machine having sent the transaction packet, the outbound game payload enabling the gaming machine having sent the transaction packet to complete the game transaction and wherein the at least one gaming machine is configured such that a first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine.

Mockapetris is relied on, in the outstanding Office Action, for its teaching of a multicast algorithm and Nguyen is relied on for its alleged teaching of a multicast system within a gaming

environment, in which a client terminal is a gaming machine. It is respectfully submitted that the applied combination does not teach or suggest the embodiment claimed in independent claim 1. Indeed, Mockapetris teaches broadcasting of a message from a sending host to a plurality target hosts, receiving a plurality of acknowledgments from the targeted hosts and the processing of the received acknowledgments at the sending host – see page 152 of Mockapetris.

The claimed embodiment, as amended, requires that **“a first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine.”** Mockapetris does not teach or suggest that the first-of-many arriving acknowledgments has any special significance or is effective to complete a transaction, as claimed. Indeed, Mockapetris discloses four different algorithms for reducing the cost of processing the acknowledgements from the targeted hosts.

➔ The first algorithm (see page 53 of Mockapetris) is a simulation algorithm involves the sender sending separate messages and to receive a corresponding number of acknowledgments in return. This algorithm does not teach or suggest that a first arriving acknowledgment is different, in its effect on the sending host, from latter arriving acknowledgments. The simulation algorithm also teaches a software ring of destinations, which is a system in which each recipient forwards the message to a next designated recipient and the last recipient returns an acknowledgment to the sender. It is apparent that, in this variant, only a single acknowledgment is received, and such a scheme does not teach or suggest that the first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine, as claimed.

➔ The second algorithm for acknowledgments, in Mockapetris, is the separate acknowledgment algorithm (see pages 153-154), in which a single message is broadcast to N targeted hosts and N acknowledgements are received in return. Again, there is no teaching or suggestion that a first-in-time arriving acknowledgment has any special functional significance or enables a transaction to be carried out, as expressly claimed in amended claim 1.

➔ The third acknowledgment algorithm disclosed by Mockapetris is the saturation algorithm. In this algorithm, acknowledgments are not used. Instead, a statistical analysis is made and a large number of copies of the messages are sent to each targeted host, so that each host is statistically likely to receive at least one copy of the message. Again, this algorithm does not even utilize acknowledgments, and much less teaches or suggests that a first-in-time arriving acknowledgment has any special functional significance or enables a transaction to be carried out, as expressly claimed in amended claim 1.

➔ The fourth and last acknowledgment algorithm of Mockapetris is the Negative Acknowledgment, or NACK. NACK algorithms are predicated upon the sender receiving a NACK from a targeted recipient that cannot acknowledge receipt of the sent message. Upon receiving the NACK, the sender again send the message to the NACK sending node. This algorithm also does not teach or suggest the subject matter of amended independent claim 1.

It is respectfully submitted that adding the secondary reference to Nguyen to the mix does not remedy the shortcomings of the primary reference to Mockapetris. Indeed, Nguyen teaches a redundant gaming network that includes a Data Collection Unit (DCU) that enables a gaming machine to communicate wirelessly with a server when its primary wired communication channel becomes inoperative – see paragraphs [0017], [0018] and [0019]. The DCU can also act as a store and forward node on the network, as described in paragraph [0043], which describes

the instance wherein both the wired and wireless transmission paths are unavailable and the DCU 324 stores data received from the gaming machines “until such time as the transmission path to the host server 328 is restored and the data can be transmitted.” Paragraph [0083] discloses that the DCUs can communicate with one another in a peer-to-peer network configuration. Paragraph [0089] recaps what has already been said regarding the store-and forward functionality of the DCU, and states that the DCU can act as a local download server and as a cache, to enable the DCU to asynchronously download data to the gaming machines. Moreover, paragraph [0091] teaches that the DCU can be located on the gaming machine itself.

It is respectfully submitted that the applied combination does not teach or suggest that the first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine, as claimed. Nguyen does teach, in paragraph [0021] that the DCU broadcasts all messages from the server to all gaming machines – this system is referred to in Nguyen as a multi-drop system:

The DCU 124 performs the communications using a multi-drop system 204. In the multi-drop system, all messages are broadcast to all of the gaming machines connected to the DCU 124. For instance, when DCU 124 polls an individual gaming machine for information, all of the gaming machines receive the message requesting polling information (i.e., the message is broadcast to all the machines on the network). However, only the gaming machine identified in the request responds to the message. As another example, when a message is sent to an individual gaming machine from a host server, all of the gaming machines receive the message but only the addressed gaming machine will process the message.

Independent method claim 108 has been amended to include similar recitations:

**completing the game transaction, by the gaming machine, upon receipt of a first in time received the outbound game payload from one of the at least two central server, irrespective of whether and when a later in time outbound game payload is received from another one or ones of the at least two central servers.**

As such, the arguments presented above relative to amended independent claim 1 are equally applicable to amended independent claim 108, and are incorporated herein by reference, as if repeated in full.

Claim 79 has been amended as follows:

79. (Previously Presented) An online gaming system, comprising:  
a communication network;  
at least two central servers, each of the at least two servers being coupled to the network, each of the at least two central servers including a synchronization engine and  
at least one gaming machine coupled to the communication network, each of the at least one gaming machine being configured to play at least one game and to carry out a game transaction for each game played and to commit each game transaction to each of the at least two central servers by sending a separate transaction packet to each of the at least two central servers, each of the separate transaction packets sent to each of the at least two central servers including an identical inbound game payload, ~~the at least one gaming machine being configured to record a synchronization log that includes identifiers of any transactions that were not acknowledged by a non-responding one of the at least two central servers after a predetermined timeout, the synchronization log being used to subsequently send the unacknowledged transactions to the non-responding one of the at least two central servers, wherein each of the two central servers are configured such that any transaction that is not acknowledged by a non-responding one of the at least two central servers is sent directly from the synchronization engine of a responding one of the at least two central servers to the synchronization engine of the non-responding central server.~~

The applied combination of references does not teach or suggest the claimed embodiment. Indeed, Mockapetris teach the broadcasting of messages through various methods, and does not teach or suggest maintaining synchronization between central servers. Nguyen teaches that the DCU can act as a store-and-forward unit when the first and second communication paths 326, 340 are unavailable, as taught in paragraphs [0043] and [0089] (cited in outstanding Office Action). Nguyen also teaches that the DCU may form an integral part of the gaming machine. However, the Examiner is respectfully requested to examine amended claim 79 and note that it is the (at least) two central servers that are configured such that any transaction that is not acknowledged by a non-responding one of the at least two central servers is sent directly from the synchronization

engine of a responding one of the at least two central servers to the synchronization engine of the non-responding central server. For Nguyen to teach this structure and functionality, Nguyen would have to teach a plurality of host servers 328, and each of the plurality of host servers 328 would have to be taught to include a synchronization engine configured as claimed; that is, configured such that any transaction that is not acknowledged by a non-responding one of the at least two central servers is sent directly from the synchronization engine of a responding one of the at least two central servers to the synchronization engine of the non-responding central server. Nguyen, however, whether considered alone or in combination with Mockapetris, does not teach or suggest this structure or functionality. Nguyen mentions in one place (paragraph [088] that there may be one or more host servers, but never discloses any communications between such one or more host servers, and much less discloses or suggests any method or functionality that would enable one host server 328 to directly send, to another (non-responding) host server 328 a transaction received from a gaming machine. Such is simply not within the purview of Nguyen.

It is noted that Nguyen teaches that the gaming machines may also function as servers. However, even in such a configuration, the combination fails. Indeed, Nguyen at paragraph [0043] states that the DCU “may act as a local interim server” and may “store the data received from the gaming machines ... until such time as a transmission path to the host server 328 is restored and data can be transmitted.” In Nguyen, that store-and-forward functionality is invoked only when the communication paths between the DCU and the host server is severed or otherwise unavailable. However, according to the embodiment of claim 79, the synchronization engine of the responding central server directly sends the transaction to the synchronization engine of the non-responding central server, which would not be possible if the communication paths therebetween were severed or unavailable. That is, if at least one of the communication paths in Nguyen were available, the

store-and-forward functionality of the DCU 324 would not be invoked, as explicitly taught in paragraph [0043]. Note that the claim necessarily requires that the communication path(s) between respective central servers be available, even though one or more central servers are non-responding; meaning that the non-responding central server(s) has/have not acknowledged receipt of the transaction. Nguyen (whether considered alone or in combination with Mockapetris) does not even consider this eventuality (communication path(s) available, but host server 328 does not acknowledge receipt of transaction packet from gaming machine and/or DCR 324). As such the applied combination fails to teach or to suggest the embodiment defined by claim 79.

Claim 94 and its dependent claims are canceled herewith.

Claim 109 defines a related embodiment:

**committing each game transaction to each of the at least two central servers by sending a separate transaction packet to each of the at least two central servers, each of the separate transaction packets sent to each of the at least two central servers including an identical inbound game payload, and recording, in the gaming machine, a synchronization log that includes identifiers of any transactions that were not acknowledged by a non-responding one of the at least two central servers after a predetermined timeout, the synchronization log being configured to enable the gaming machine to subsequently send the unacknowledged transactions to the non-responding one of the at least two central servers.**

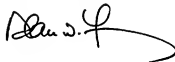
Again, the DCUs 324 are taught, in Nguyen, to only operate in store-and-forward mode when the both transmission paths are disrupted. Claim 109 specifically calls for the synchronization log (which is nowhere taught or suggested in Nguyen) to enable the gaming machine to subsequently send unacknowledged transactions to one or more non-responding ones of the central servers, which Nguyen does not teach or suggest, even in combination with Mockapetris. Kindly note that, according to the claimed embodiments, a central server may be busy or otherwise unavailable or unable to respond, even when the communication paths between the gaming machines and the central servers are intact, which is an eventuality that is not even

addressed by Nguyen. Moreover, note that Nguyen teach that each gaming machine communicates with only a single host server, whether directly or through a DCU324. There is not teaching or suggestion in the applied combination of a gaming machine, through a synchronization log or by other means, sending unacknowledged transactions to more than one host server 328. Claim 109, therefore, is not believed to be either taught or suggested by Nguyen alone or in combination with Mockapetris.

Claim 110 is canceled

Applicants' attorney believes that the present application is now in condition for allowance and passage to issue. If any unresolved issues remain, the Examiner is respectfully invited to contact the undersigned attorney of record at the telephone number indicated below, and whatever is required will be done at once.

Respectfully submitted,



Date: January 22, 2008

By: \_\_\_\_\_

Alan W. Young  
Attorney for Applicants  
Registration No. 37,970

YOUNG LAW FIRM, P.C.  
4370 Alpine Rd., Ste. 106  
Portola Valley, CA 94028  
Tel.: (650) 851-7210  
Fax: (650) 851-7232

C:\YLF\CLIENTS\JMG\5872 (Univ Gaming Server)\5872 AMEND.5.doc